

Adaptation and Coaching of Periodic Motion Primitives through Physical and Visual Interaction

Andrej Gams^a, Tadej Petrič^{a,d}, Martin Do^b, Bojan Nemec^a, Jun Morimoto^c,
Tamim Asfour^b and Aleš Ude^{a,c}

^a *Humanoid and Cognitive Robotics Lab,
Department for Automatic, Biocybernetics and Robotics, Jožef Stefan Institute,
Jamova cesta 39, 1000 Ljubljana, Slovenia*

^b *High Performance Humanoid Technologies Lab,
Institute of Anthropomatics and Robotics, Karlsruhe Institute of Technology
Adenauerring 2, 76131, Karlsruhe, Germany*

^c *Department of Brain Robot Interface, ATR Computational Neuroscience Labs
Kyoto 619-0288, Japan*

^d *Biorobotics Laboratory, École Polytechnique Fédérale de Lausanne
Station 14, CH-1015 Lausanne, Vaud, Switzerland e-mail: {andrej.gams, tadej.petric,
bojan.nemec, ales.ude}@ijs.si,
{martin.do, asfour}@kit.edu
xmorimo@atr.jp*

Abstract

In this paper we propose and evaluate a control system to 1) learn and 2) adapt robot motion for continuous non-rigid contact with the environment. We present the approach in the context of wiping surfaces with robots. Our approach is based on learning by demonstration. First an initial periodic motion, covering the essence of the wiping task, is transferred from a human to a robot. The system extracts and learns one period of motion. Once the user/demonstrator is content with the motion, the robot seeks and establishes contact with a given surface, maintaining a predefined force of contact through force feedback. The shape of the surface is encoded for the complete period of motion, but the robot can adapt to a different surface, perturbations or obstacles. The novelty stems from the fact that the

feedforward component is learned and encoded in a dynamic movement primitive. By using the feedforward component, the feedback component is greatly reduced if not completely canceled. Finally, if the user is not satisfied with the periodic pattern, he/she can change parts of motion through predefined gestures or through physical contact in a manner of a tutor or a coach.

The complete system thus allows not only a transfer of motion, but a transfer of motion with matching correspondences, i. e. wiping motion is constrained to maintain physical contact with the surface to be wiped. The interface for both learning and adaptation is simple and intuitive and allows for fast and reliable knowledge transfer to the robot.

Simulated and real world results in the application domain of wiping a surface are presented on three different robotic platforms. Results of the three robotic platforms, namely a 7 degree-of-freedom Kuka LWR-4 robot, the ARMAR-IIIa humanoid platform and the Sarcos CB-i humanoid robot, depict different methods of adaptation to the environment and coaching.

Keywords: dynamic movement primitives, force control, coaching, human-robot interaction

1. Introduction

Learning by demonstration, as an approach of acquiring trajectories in robotics [1], can only be effective if it enables adaptation of the demonstrated policy to the current situation of the task or the environment [2]. For example, when learning a wiping behavior, which is a rather trivial skill for humans, the robot must acquire the correct characteristics of motion, but must also maintain contact with the surface it is wiping. Such skill transfer from a human to a robot, where not only the

motion but also the constraints imposed by the task are important, is the motivation behind this paper. We propose a system that enables a robot to learn actions which require continuous non-rigid contact with the environment through human demonstrations and interactive coaching. The coaching mechanisms enable a human teacher to efficiently guide the robot towards a goal-directed execution.

Learning by demonstration often exploits the means of encoding the motion characteristics of an action by generalizing demonstrated trajectories from the performing subject and the current situation. Different approaches exist, for example splines and wavelets [3, 4], which are effective for imitation learning, but do not allow easy online modulation. Another option are Gaussian Mixture Regression [5] and Gaussian Mixture Models [6, 7], used to estimate the entire attractor landscape of a movement skill from several demonstrations. To ensure stability of the dynamical system toward an attractor point, a constraint optimization problem in a nonconvex optimization landscape needs to be solved. Yet another option is the use of Hidden Markov Models [8]. Different dynamical systems can also be used.

Another type of dynamical systems are dynamic movement primitives (DMPs) [9], which focus on the representation of single movements by a set of differential equations. A DMP can represent a control policy in a compact way and its attractor landscape can be adapted by only changing a few parameters. Compared to representations proposed in [6, 7], only a simple system of linear equations need to be solved. DMPs can be used for representing classes of movements using statistical learning techniques [2, 10], for combining trajectories in a dynamic way [11, 12], and for reinforcement learning [13, 14, 15, 16]. In this paper we exploit the DMP framework to enable continuous non-rigid contact with the environment, based on force feedback.

Adaptation of learned trajectories to external feedback was previously discussed in different settings and applications, using different trajectory representations. The use of force feedback to learn and improve task execution was widely considered in robotics, see for example the book chapter by Villani and De Schutter [17]. One of the best known approaches is the method proposed by Hogan [18], where force feedback is used to change the output velocity of a manipulator. This technology is the basis for the DMP adaptation proposed in this paper.

DMPs themselves were already used for adaptation to forces. In [19] the authors used an interaction force and the parallel force/position control law to modulate the velocity of the dynamical system. Pastor et al. [20, 21] have also combined force controllers and DMPs in an approach for modifying DMPs at the acceleration level, allowing for reactive and compliant behaviors. They used the demonstrated trajectory profiles as reference, while [22] applied reinforcement learning to further optimize the behavior. A modulation approach at the acceleration level of a DMP for physically coupled dual-agent tasks was reported by Kulvicius et. al [23], but the learning was applied to acquire appropriate feedback gains instead of reference trajectories. On the other hand, Gams et al. [24] utilized coupled DMPs with force feedback at the velocity level. Combined with iterative learning control, their approach can achieve the desired force interaction for rigid contacts. Iteratively approaching a desired behavior has been applied for in some programming by demonstration approaches. For example, Sauser et al. [25] showed grasp adaptation through human corrections, while Calinon and Billard [26] showed gesture learning. On the other hand, iteratively approaching a desired behavior was also shown in combination with DMPs in a peg-in-hole task [27], where reference force-torque profiles were used as means for autonomously

improving the execution performance. In this work the force controller was not applied within the DMP framework. Haptic feedback for improving the teacher demonstration was also used by Rozo et al. [28], who addressed the problem of what to imitate based on the mutual information between perceptions and actions. HMMs and GMR were used to encode the demonstrations and for the robotic execution of the learned tasks. The method was augmented in order to be applicable also for the task of pouring [29]. Adaptation of trajectories is not limited to one-arm behaviors. An approach for bimanual operation based on dynamical systems by adding local corrective terms was discussed by Calinon et al. [30].

In this paper we consider the transfer of skills from a human to a robot through coaching. The transfer is not limited to the motion, but includes the execution of the task in contact with the environment. We consider two problems of on-line motion adaptation for the actual completion of the task. The first is the adaptation to the external environment in order to achieve desired forces of non-rigid contact throughout the complete trajectory. The second is adapting the trajectories to the interventions of an instructor, modifying the trajectories through physical contact or with the use of predefined gestures. The interaction puts the instructor into the role of a tutor who coaches the robot to achieve the desired performance. Both adaptation to the environment and coaching rely on the use of a unified trajectory representation, i. e. the dynamic movement primitives (DMPs). The combination creates an intuitive and user-friendly interface to learning and modifying robotic trajectories with the potential of creating complex object-interaction trajectories.

Not many papers describe adaptation of learned trajectories for non-rigid contacts. Initial results of DMP adaptation methods, expanded on in this paper, were presented in [31, 32]. The approach was expanded on by Ernesti et al. [33] to

include transient motions and [34] to include structural bootstrapping from experience. Wiping with a robot has also been studied from other perspectives, including using dynamic models and operational space dynamics [35].

Coaching has been applied also in context of other robotic tasks. Gruebler et al. [36] used voice commands as a reward function in their learning algorithm. Verbal instructions of non-experts were used to modify movements obtained by human demonstration [37]. Physical contact was also used, for example, by Lee & Ott [38] who used kinesthetic teaching with iterative updates to modify the behavior of a humanoid robot. Coaching based on gestures and obstacle avoidance algorithms was applied to DMPs [39]. This approach is expanded on in this paper with force feedback.

In the next section we provide the basics of DMPs and the algorithm of encoding them. Section 3 provides the core algorithm of the adaptation approach. Three different methods are explained. Coaching, as the means to adapt parts of the trajectory based on the user input is explained in Section 4, followed by the results in Section 5 and a discussion with conclusions.

2. Learning of Periodic Dynamic Movement Primitives

In this paper we build on periodic dynamic movement primitives. For the sake of completeness we provide the basics of the DMP notation and an algorithm for extracting the frequency of the demonstrated signals. The algorithm of learning of weights that encode a DMP follows. It is the basis for both adaptation to external force and the coaching algorithms.

2.1. Periodic DMPs

The formulation of DMPs in this paper is based on [2]. For a complete DMP overview see [9]. The description is for clarity limited to a single degree of freedom (DOF), i. e. one of the external task-space coordinates, denoted by y . Temporally scaled velocity is denoted by z . Note that DMPs can be applied to joint space coordinates as well. y and z should not be mistaken with the axes of a coordinate system, which are in this paper denoted by x_p, y_p, z_p .

A nonlinear system of differential equations that defines a periodic DMP is given by

$$\dot{z} = \Omega(\alpha_z(\beta_z(g - y) - z) + f(\phi)), \quad (1)$$

$$\dot{y} = \Omega z. \quad (2)$$

The nonlinear part of (1), $f(\phi)$, known as the forcing term, is comprised of a linear combination of N radial basis functions $\Gamma_i(\phi)$

$$f(\phi) = \frac{\sum_{i=1}^N w_i \Gamma_i(\phi)}{\sum_{i=1}^N \Gamma_i(\phi)} r. \quad (3)$$

Radial basis functions $\Gamma_i(\phi)$ are defined by

$$\Gamma_i(\phi) = \exp(h_i(\cos(\phi - c_i) - 1)). \quad (4)$$

Parameter r is the amplitude control parameter, $h_i > 0$ are the widths of the kernels and c_i spreads them equally along the phase ϕ from 0 to 2π in N steps. The parameters $\alpha_z, \beta_z, > 0$ and $\alpha_z = 4\beta_z$ make the system (1) – (2) critically damped. The system oscillates as given by $f(x)$ around the goal g . To realize multiple DOFs we use separate sets of (1) – (2), and a single canonical system given by (5) to synchronize them through the common phase.

The phase variable ϕ provides the indirect dependency on time. It can increase with constant rate, where the parameter Ω denotes the frequency

$$\dot{\phi} = \Omega. \quad (5)$$

When learning the frequency does not have to remain constant, but needs to be estimated, for example with adaptive frequency oscillators as in [40] and [41]. In our case we used the system proposed in [41] for online extraction of frequency of motion and to encode one period of motion with the weights w_i , $i = 1, \dots, N$, where N is the number of kernel functions. The frequency estimation is based on a feedback structure containing an adaptive frequency oscillator and an adaptive Fourier series.

The adaptive frequency oscillator is governed by the following feedback structure

$$\dot{\phi} = \Omega - Ke_o \sin \phi, \quad (6)$$

$$\dot{\Omega} = -Ke_o \sin \phi, \quad (7)$$

$$e_o = y_{\text{demo}} - \hat{y}, \quad (8)$$

where K is the coupling strength, ϕ is the phase of the oscillator, e_o is the input into the oscillator and y_{demo} is the input signal. The feedback loop signal \hat{y} in (8) is provided by the Fourier series

$$\hat{y} = \sum_{a=0}^M (\alpha_a \cos(a\phi) + \beta_a \sin(a\phi)). \quad (9)$$

Here M is the number of components of the dynamic Fourier series and α_a , β_a are the amplitudes associated with the series. They are estimated as follows:

$$\dot{\alpha}_a = \eta \cos(a\phi) e_o, \quad (10)$$

$$\dot{\beta}_a = \eta \sin(a\phi) e_o, \quad (11)$$

where η is the learning constant and $a = 0, \dots, M$.

2.2. Learning of DMPs

To encode a periodic trajectory as a DMP, we need to determine the duration of one period of motion, for example by using the above-described adaptive frequency oscillators. Once the frequency of the demonstrated motion is established, we need to learn the weights of the DMP to encode the shape of the demonstrated motion. The latter is accomplished using incremental locally weighted regression (ILWR) [42]. The target data for fitting is constructed from the demonstration trajectory y_{demo} , which is the desired trajectory of motion. The target function for fitting, originating from (1)-(2) is therefore

$$f_{\text{targ}} = \frac{1}{\Omega^2} \ddot{y}_{\text{demo}} - \alpha_z \left(\beta_z (g - y_{\text{demo}}) - \frac{1}{\Omega} \dot{y}_{\text{demo}} \right). \quad (12)$$

It is obtained by matching y from (1) – (2) to y_{demo} , z to $\dot{y}_{\text{demo}}/\Omega$, and \dot{z} to $\ddot{y}_{\text{demo}}/\Omega$. This means that we basically learn how to force the otherwise critically damped spring-mass system given by the linear part of (1) – (2) to follow the desired trajectory.

Given f_{targ} , w_i is updated incrementally for each time-step j as

$$w_{i,j+1} = w_{i,j} + \Gamma_{i,j+1} P_{i,j+1} r e_j \quad (13)$$

$$P_{i,j+1} = \frac{1}{\lambda} \left(P_{i,j} - \frac{P_{i,j}^2 r^2}{\Gamma_i + P_{i,j} r^2} \right) \quad (14)$$

$$e_j = f_{\text{targ},j} - w_{i,j} r. \quad (15)$$

Γ_i are the kernel functions. P_i , in general, is the inverse covariance of w_i [43]. The recursion is started with $w_i = 0$ and $P_i = 1$. r is the amplitude gain. The forgetting factor is defined by $\lambda \leq 1$. Useful range of λ is between 0.97 and 1. If $\lambda < 1$,

then the incremental regression gives more weight to recent data, meaning that it tends to forget older ones.

3. Adaptation to Environment

Adaptation to the environment, as proposed in this paper, assumes that the environment can not change rapidly, i. e. an object, such as a table or a kitchen sink, does not rapidly change shape or height. The setting of the environment, on the other hand, can be completely arbitrary. This assumption allows gradual adaptation of motion through learning, and is the basis of the proposed algorithm. If the environment does not change rapidly, then a correct reference of motion (if followed) will achieve the desired behavior. The referential trajectory is the output of the DMP, and can be interpreted as a feed-forward control signal. This is augmented with the feedback control loop for instantaneous reaction, and to allow gradual adaptation. The use of the learned feed-forward component (the output of the DMP) reduces the need for feedback adaptation, which allows for greater accuracy. Furthermore, the use of the DMP allows for standard DMP features, such as easy modulation with the change of only a few parameters.

In this paper we propose two means of applying force feedback to change the output of the DMP, i. e. the feed-forward component of the control signal. The first is in changing the reference for learning a DMP. The second is in bootstrapping the force signal directly into the DMP weight adaptation.

3.1. Changing the Reference

The original use of DMPs allows the encoding of demonstrated trajectories for imitation, i. e., the demonstrated trajectory is the reference. If the reference is changing over time, so is the output of the DMP. In this algorithm we exploit

force feedback to change the reference of the DMP, i. e. y_{demo} . The change of the reference trajectory occurs through the change of the end-effector velocity as a function of force, known as the velocity-resolved approach [17]

$$\mathbf{v}_r = \mathbf{S}_v \mathbf{v}_v + \mathbf{S}_F (\mathbf{K}_i \mathbf{e}_f + \mathbf{K}_p \dot{\mathbf{e}}_f), \quad (16)$$

$$\mathbf{e}_f = \mathbf{F}_0 - \mathbf{F}_m. \quad (17)$$

The variable \mathbf{v}_r stands for the resolved velocities vector, \mathbf{S}_v for the velocity selection matrix, \mathbf{v}_v for the desired velocities vector, $\mathbf{K}_i, \mathbf{K}_p$ for the force gain matrices, \mathbf{S}_F for the force selection matrix, \mathbf{F}_m for the measured force and \mathbf{F}_0 the desired force of contact. Essentially, the selection matrices \mathbf{S}_v and \mathbf{S}_F determine which directions of motion are affected by the force. They are determined by the user, who knows which directions of motion need to be altered.

To get the desired positions we use

$$\mathbf{y}_r = \mathbf{y}_{\text{demo}} + \mathbf{S}_F \left(\int \mathbf{v}_r dt \right) = \mathbf{y}_{\text{demo}} + \mathbf{S}_F \left(\int \mathbf{K}_i (\mathbf{F}_0 - \mathbf{F}_m) dt + \mathbf{K}_p (\mathbf{F}_0 - \mathbf{F}_m) \right). \quad (18)$$

Here \mathbf{y}_r is the resolved position (and possibly also orientation) of the robot, taking the place of \mathbf{y}_{demo} . We see in (18) that \mathbf{y}_{demo} has both an integral and a proportional feedback loop, with \mathbf{K}_p being the proportional gain. Combining the two allows for zero steady-state error within the integral loop and faster reactions to possible unforeseen perturbations within the proportional loop.

When wiping a flat horizontal surface, such as an average table, (16) – (18) become less complex. In this case the robot needs to establish contact in a vertical direction, typically z . We obtain: $\mathbf{S}_v = 0$, $\mathbf{K}_i = \text{diag}(0, 0, k_i, 0, 0, 0)$, $\mathbf{K}_p = \text{diag}(0, 0, k_p, 0, 0, 0)$, $\mathbf{S}_F = \text{diag}(0, 0, 1, 0, 0, 0)$. Only the desired end-effector height

z_p is modified in each discrete time step Δt , and (16) becomes (19).

$$\dot{z}_p(t) = k_i e_f(t) + k_p \dot{e}_f(t), \quad (19)$$

$$e_f(t) = F_0(t) - F_z(t). \quad (20)$$

Taking into account the initial condition and numerical integration, it results in (21)

$$z_p(t) = z_{p0} + k_i e_f(t) \Delta t + k_p e_f(t). \quad (21)$$

Here z_{p0} is the initial z_p value (starting height), k_i and k_p are positive constants for force gains, F_z is the measured force in the z_p direction and F_0 is the desired force of contact. The movement is constant in $-z_p$ direction when there is no contact, or maintains contact force F_0 when an object is encountered.

The learning of the DMP in the direction that is being modified by force, for example in z_p direction as shown in (19) – (21), is done by modifying the weights w_i for the selected DOF (determined by \mathbf{S}_F) in every time-step by using incremental locally weighted regression as given by (13) – (15). The *demonstrated* trajectory is being constantly modified by the force feedback and therefore the DMP weights are constantly re-evaluated until a steady-state is reached. Since this approach uses the position of the end-effector as input, and not the force, it has no difficulties with the noisy measured force signal.

The admittance control scheme given by (16) – (18) is subject to the gains \mathbf{K}_i and \mathbf{K}_p . High gains will result in fast reactions when encountering a force. At the same time, being subject to time discretization, specifically at low sampling rates, too high gains may produce instabilities. A trade-off has to be made based on the desired behavior. We empirically set \mathbf{K}_i and \mathbf{K}_p values and also limited the force feedback to a maximum absolute value.

3.2. Direct Adaptation of the DMP

The change of y_{demo} will inherently cause some delay typical for feedback controllers. To cancel the delay of the algorithm that changes the reference for learning, we exploit the incremental weight fitting algorithm (13) – (15) for learning of periodic DMPs. The basic idea here is that we replace the error signal for weight fitting associated with imitation with a different signal, for example the difference between the measured and desired forces in force interaction.

Let's assume that a given trajectory is encoded as a DMP with weights \mathbf{w} . The trajectory follows the demonstrated trajectory if the error signal in (15) is equal to $e_j = 0$, meaning that \mathbf{w} does not change. We now replace the imitation-related error signal (15) with a force-dependent term

$$e_j = k_l(F_0 - F_m). \quad (22)$$

By using e_j from (22) in (13), the weights of the DMP will be updated whenever the measured and the desired forces are different. Therefore it will adapt the trajectory to fulfil the condition of (22), which is that the actual force of contact F_m is the same as the desired force F_0 in the given direction. Parameter k_l is a positive constant, determined empirically. Note that the implementation of adaptation should take care that the values of the inverse covariance P_i do not decrease to $P_i \cong 0$ as this will stop the adaptation, given that the update of weights is multiplied by P .

A feedback term can also be added to the acceleration level of the DMP for instantaneous reaction, changing (1) into

$$\dot{z} = \Omega(\alpha_z(\beta_z(g - y) - z) + f(\phi) + d(F)). \quad (23)$$

The feedback term can be a simple proportional control law with gain $k_{fb} > 0$, for example $d(F) = k_{fb}(F_0 - F)$. In this paper we name the trajectory adaptation method based on (22) the *Direct* method.

In simulation, where we can model the forces of contact with displacement of the elastic environment with stiffness k_{env} , we can rewrite (22) into $F = k_{env}(y_0 - y) = k_{env}\bar{y}$. Any difference of forces at end-effector will therefore introduce a position difference $k_l k_{env}(y_0 - y)$, which will through (13) reflect in $f(\phi)$. From (1) – (2) we can see that through integration of the DMP differential equations, $f(\phi)$ (and consequently $y_0 - y$) is integrated twice, which results in a slight delay.

From a physical standpoint, the linear part of (1) represents accelerations of a spring-mass system, while $f(\phi)$ provides the modification for accelerations that force the system to follow the desired trajectory (hence earning the name *forcing* term). In order to exclude the above mentioned delay from position-difference integration, we need to change (22) so that it provides proper accelerations for the second order DMP spring-mass system. These are calculated according to (12). We therefore write

$$e_j = \frac{1}{\Omega^2} k_2 \ddot{y} - \alpha_z \left(\beta_z (g - k_2 \bar{y}) - \frac{1}{\Omega} k_2 \dot{y} \right), \quad (24)$$

where $k_2 \bar{y} = k_l k_{env}(y_0 - y)$ models the forces. In this paper we name the trajectory adaptation method based on error signal (24) the *Derived* method.

Table 1 provides the basic characteristics and differences of the three methods described in this Section.

	Chg. Ref.	Direct	Derived
Property	Velocity-resolved approach (3.1) changes the reference of DMP learning	Error of DMP learning (3.2) is defined with the error of force tracking	Error of DMP learning is defined with the error of force tracking, modified into accelerations of the DMP spring-mass system
Advantage	Classical velocity resolved approach with well known stability properties and behavior	Simple; no additional parts of the algorithm; core of the coaching algorithm.	Completely reduces the error
Drawback	Subject to delay due to the integral part of the force controller	Will not completely cancel out the error due to the delay of integration	Subject to noise of the derivation that modifies the error of force tracking

Table 1: Properties, advantages and drawback of the variations of the DMP adaptation to the environment.

4. Coaching

During trajectory learning the demonstrator repeats several periods of motion and the collected data are given as reference to the incremental locally weighted regression. The trajectory is learned, but it might not exactly perform the desired task, as is often the case when giving instructions to another person on how to perform something. When the resulting motion is not satisfactory, the demonstrator can coach the other person, specifying how to alter the motion in certain parts, or simply showing the complete motion again.

In order to avoid re-learning of the complete trajectory, we can exploit the same mechanism as was applied for the *Direct* method to change only parts of the trajectory. We again rely on changing (15). If $e_j = 0$, there is no learning and the robot just repeats the trajectory it learned during the demonstration. Again, for a single degree of freedom, we change (15) into

$$e_j = C(\text{input}), \quad (25)$$

making the error a function of the input, where input can be either the force applied to the robot or the demonstrator's pointing gesture, visually illustrating in which direction to change the trajectory. For the case of force input, (25) changes into

$$e_j = k_l F, \quad (26)$$

where parameter k_l scales the measured force F . The measured force in this case should be the force exerted by the coach on the robot. If the robot is in contact with an object, for example when wiping the table, one must distinguish between the forces that arise from the contact with the table and as a result of friction, and the forces applied by the human operator. A simple solution is to decouple forces by direction.

Pointing gestures can be used instead of the force. We used active motion capture markers to first demonstrate a motion and later use the same markers and their relative positions for tutoring. We defined the following repulsive force field

$$e_j(x) = \begin{cases} 0 & p > 0.1 \\ (0.001/p^2 - 0.1)/40 & p \leq 0.1, p_{1z} > p_{2z} \\ (-0.001/p^2 - 0.1)/40 & \text{otherwise} \end{cases} \quad (27)$$

where p stands for the distance between the robot and the closest marker attached to the coach's hand. Index i_z is the z_p axis location of the i -th marker. The given force field has no effect on the robot if the closest marker is more than 10 cm away, whereas its effect increases quadratically with proximity, effectively pushing the robot away if $p \approx 0$. The relative location of the markers also defines if the robot is being pushed away or pulled towards the tutor. The given force field was determined empirically.

The design of the force field has a direct impact on the behavior of the robot. Force fields have previously been applied to DMPs for obstacle avoidance [44]. The same field can be used for coaching. In this case we coach the robot through predefined gestures as depicted in Fig. 1.

A 3-DOF DMP is defined by

$$\dot{\mathbf{z}} = \Omega(\alpha_z(\beta_z(\mathbf{g} - \mathbf{y}) - \mathbf{z}) + \mathbf{C}_y + \mathbf{f}), \quad (28)$$

where \mathbf{y} , \mathbf{z} , \mathbf{g} , \mathbf{C}_y , and \mathbf{f} are three dimensional values (for positions, additional dimensions can be added for orientations). The definition of the coupling term \mathbf{C}_y prescribes the behavior of the robot. We designed the coupling term as a modified obstacle avoidance coupling term \mathbf{C}_y from [44], now given by

$$\mathbf{C}_y = \gamma s(\|\mathbf{o} - \mathbf{x}\|) \exp(-\beta\phi) \mathbf{d}. \quad (29)$$

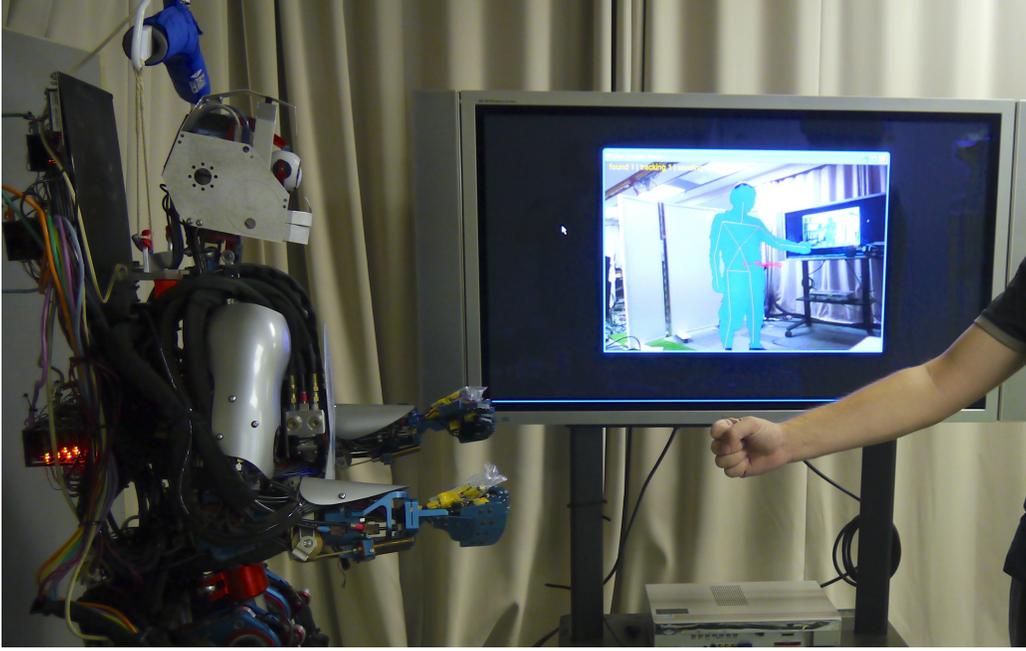


Figure 1: Experimental setup for coaching of periodic motion on the Sarcos CBi humanoid robot using predefined gestures. A Kinect RGB-D camera detected the posture of the human next to the robot. The position and the choice of the arm (left or right) determined the coaching behavior.

Here \mathbf{x} is the Cartesian position of the end-effector, \mathbf{o} is the center position of the perturbation potential field (defined by hand position), \mathbf{d} is the perturbation direction (defined by the pointing gesture), γ and β are the scaling factors, ϕ is given by

$$\phi = \arccos \left(\frac{(\mathbf{o} - \mathbf{x})^T \dot{\mathbf{x}}}{\|(\mathbf{o} - \mathbf{x})\| \|\dot{\mathbf{x}}\|} \right). \quad (30)$$

$s(r)$ is defined as

$$s(r) = \frac{1}{1 + e^{\eta(r-r_m)}}, \quad (31)$$

where η is the scaling factor and r_m the distance at which the perturbation field should start affecting the robot's motion.

5. Results

In this section we discuss simulated and real-world results of the adaptation to the environment and the coaching.

5.1. Simulated results

We first present the results of a comparison of the three possibilities of adapting to the environment, namely by changing the reference trajectory and the two methods of DMP adaptation - the Direct and the Derived methods.

The simulated experiment was designed to show adaptation of all three methods to a tilted flat surface. In the top plot of Fig. 2 we can see red line depicting the reference, i. e. the table. As it is tilted and the robot is moving left-right, it is a saw-signal. The three output signals of the adaptation are also shown. The green line depicts the trajectory when using the approach of changing the reference for learning, as given by (16) – (18). Note that there is some delay in the adaptation as a consequence of the velocity-resolved force control approach of changing the reference. The bottom plot shows that the error does not completely disappear, but is reduced. The values of \mathbf{K}_i , \mathbf{K}_p were determined empirically.

The *Direct* and the *Derived* methods also need some time to adapt but considerably reduce the error in the steady state. We can see that the derived method, given by (24) and depicted in blue, completely cancels out the error, unlike the direct method, which is given by (22) and is depicted in black. This is because the transformation of the error signal is in fact an inverse of the DMP itself and the adaptation is therefore linear at the output, while the direct method uses a second order DMP system that receives linear correction signals. As stated in Table 1, the derived method utilizes first and second order derivatives of the error signal,

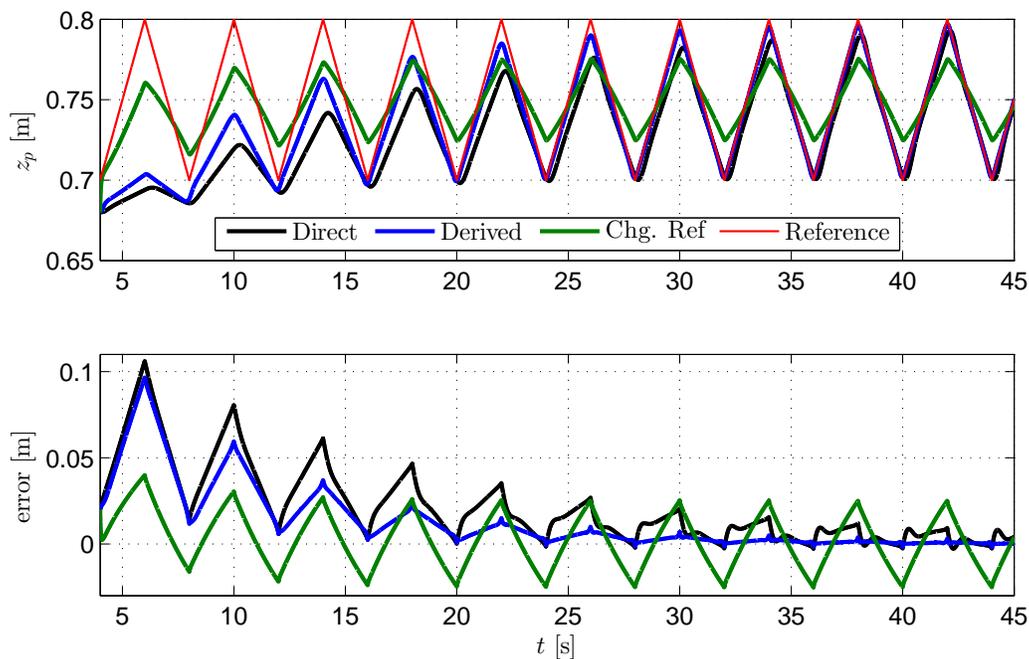


Figure 2: The results of simulated trajectory adaptation using three different control methods, with a tilted flat surface as a reference. The experiment started with the robot already in contact with the surface. We can see the reference (red) and the three resulting trajectories in the top plot. The errors of adaptation are shown in the bottom plot. See the text for a description of separate lines.

which could prove extremely noisy.

5.2. Adaptation to Environment

This task was performed using a Kuka 7 degree-of-freedom LWR-4 robot, controlled at 500Hz through Matlab Simulink. The wiping motion was first transferred from a human to a robot using an Optitrack motion capture system with markers on the sponge. The recorded task-space motion was reproduced by the robot while the method of changing the reference, given by (18), ensured that the robot achieved the contact with a surface needed for effective wiping. Figure 3

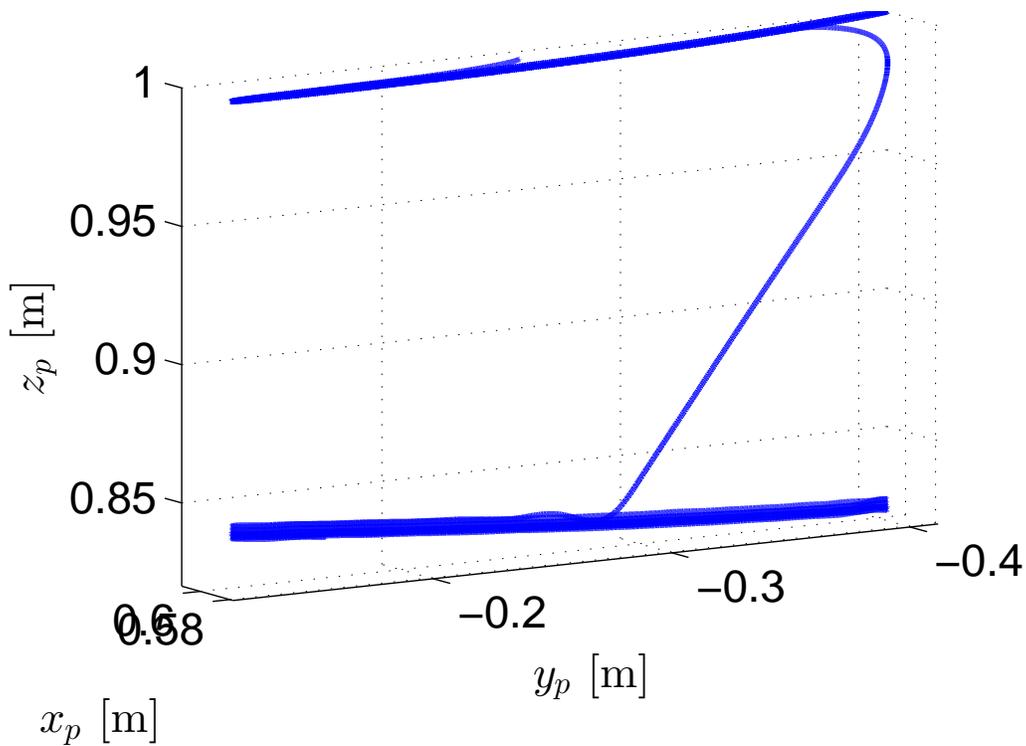


Figure 3: The complete 3-D trajectory resulting from the adaptation of the demonstrated trajectory in p_z direction using the approach of changing the reference. The force results are depicted in Fig. 5.

shows the 3-D trajectory resulting from an adaptation of the demonstrated motion to a flat, yet slightly tilted surface. The tilting angle was set completely arbitrarily and was not measured.

The force feedback signal for adaptation to a flat, horizontal surface is shown in Fig. 4. The top plot shows the trajectory in the z_p direction. Once the adaptation has started, the robot approaches the table at a finite speed, which was limited beforehand. The bottom plot of Fig. 4 shows the forces. Some oscillations are the result of friction from dragging the sponge left-right during the wiping.

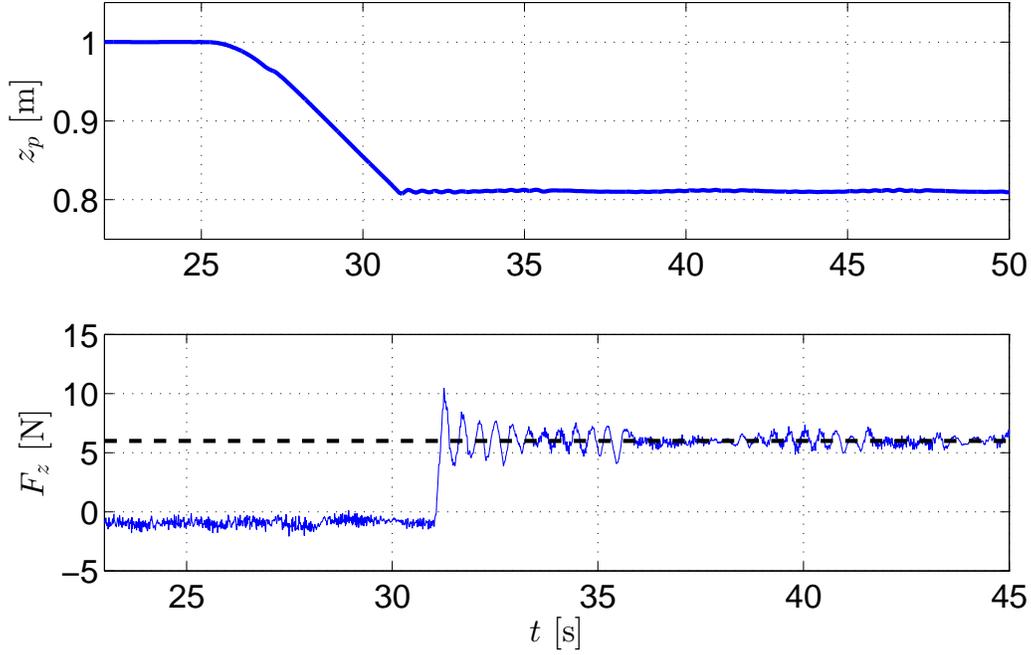


Figure 4: The top plot shows real world results of adaptation of motion in p_z direction (downwards). The resulting forces with the referential contact force set to 6 N (dashed line) is shown in the bottom plot. As the robot performed left-right wiping motion, some oscillations due to the contact are visible in the force measurement.

Figure 5 shows the results of wiping a slightly tilted surface, with the z_p trajectory in the top plot and the force profile in the bottom plot. Notice the hysteresis of resulting forces, which shows that adaptation takes some time. The integral part of adaptation in (18) introduces delays, which cause these force oscillations. Just as in the simulated environment, the gains \mathbf{K}_i , \mathbf{K}_p determine the behavior of the robot.

The real-world wiping experiment with different, arbitrarily tiled flat surfaces and a curved surface is shown in Fig. 6. All experiments on the Kuka LWR-4

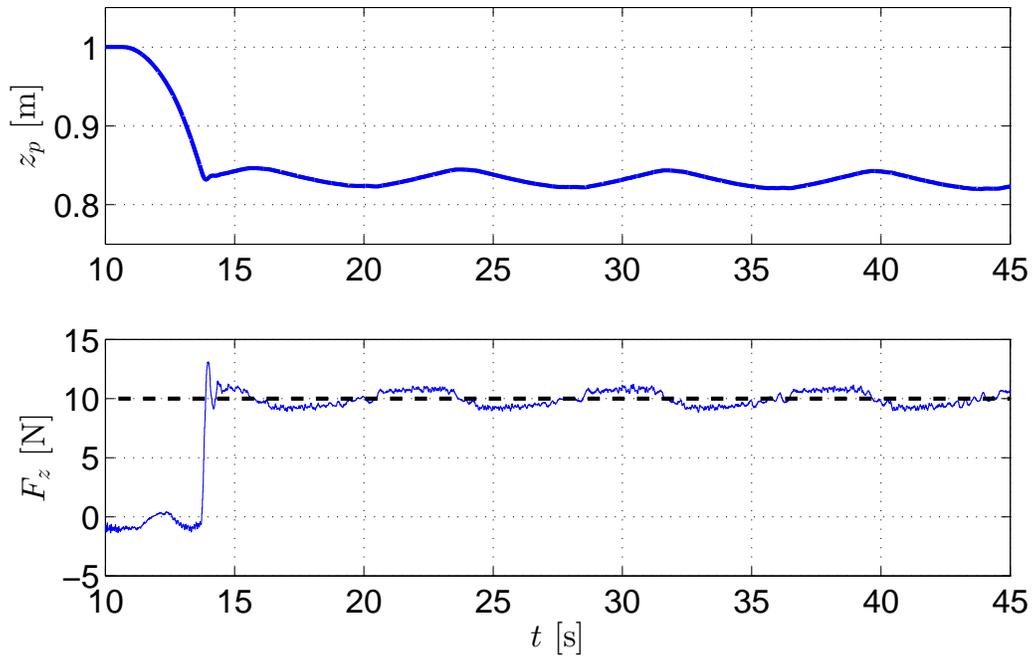


Figure 5: The trajectory of motion when adapting to a flat but tilted surface in the top plot. The resulting forces show a clear hysteresis resulting from moving up or down the slope in the bottom plot. The oscillations in the force plot are a result of the delay of adaptation, caused by the integral part of the adaptation in (18).

robot are depicted in the accompanying video.

We also implemented the direct method, given by (22). In this scenario the robot was already in contact with the surface and the reference was a sinusoidal force trajectory. Fig. 7 shows the results. A low value of k_I was used in (22) for safety. A higher value would reduce the time needed for adaptation, but a too high value would make the contact unstable. The value used was determined empirically.

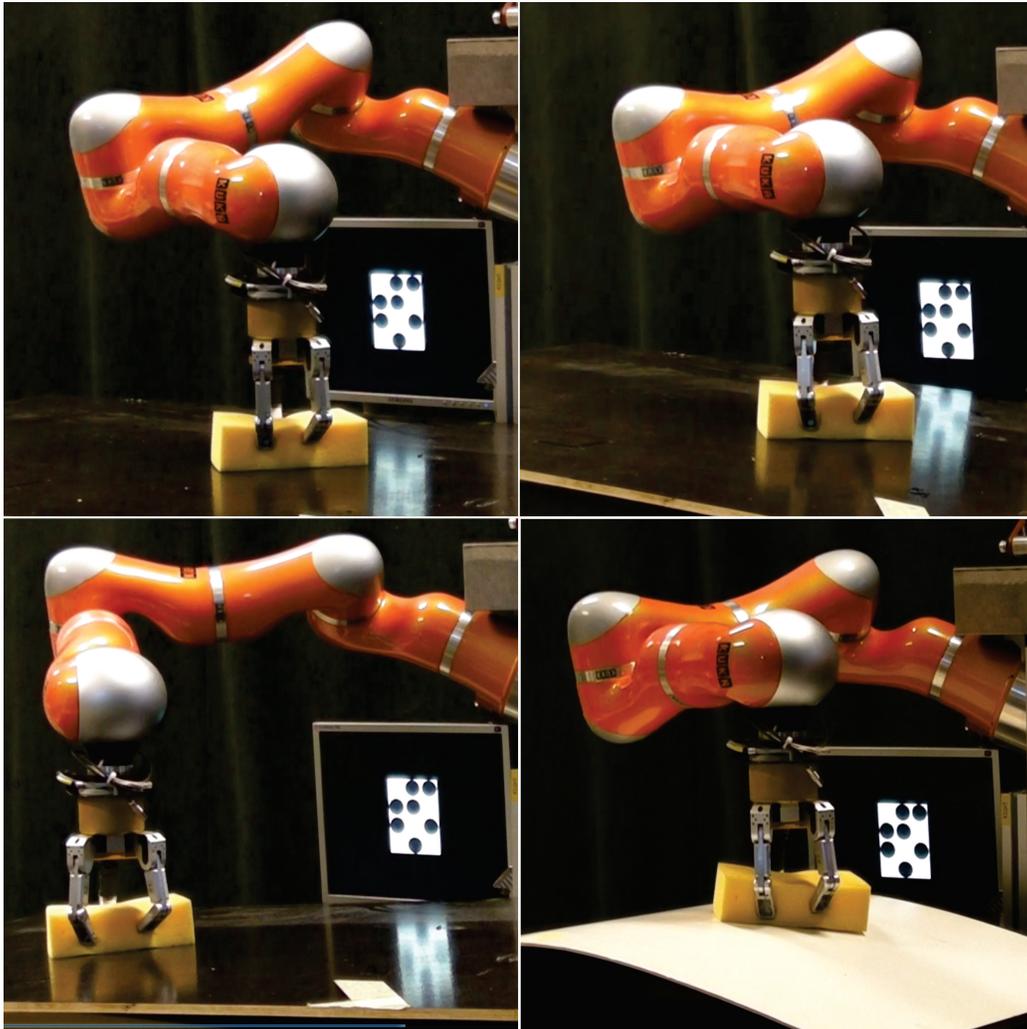


Figure 6: Kuka LWR-4 7DOF robot wiping differently tilted surfaces and a curved surface. Top left: horizontal surface. Top right: right-tilted surface. Bottom left: left-tilted surface. Bottom right: curved surface. Kuka LWR-4 robot experiments are also depicted in the accompanying video.

5.3. Adaptation to the Environment and Coaching

In this section we show the results of changing only a part of the trajectory using coaching. Results using force interaction are presented first, followed by

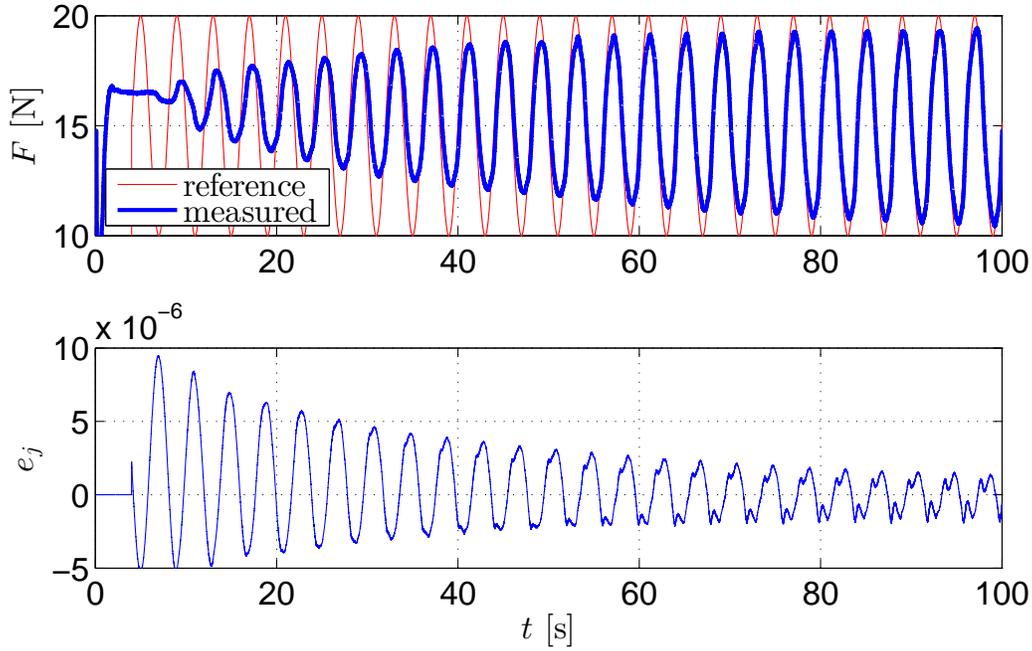


Figure 7: The results of adapting the robot trajectory using the direct method, with a sinusoidal referential force. The experiment started with the robot already in contact with the surface. The referential and resulting forces are in the top plot, while the error signal, given by (22) is in the bottom plot.

results based on coaching gestures.

Fig. 8 shows the robot end-effector trajectory before and after coaching. The initial robot wiping motion is in green. The blue line shows the trajectory of the robot during coaching, i. e. while the human was pushing/pulling on it. The measured contact forces are shown in Fig. 9. Four clear peaks of force show where the human pushed/pulled on the robot. The final wiping motion of the robot after coaching is shown in red. The initial motion was performed using a previously learned DMP, the one from Fig. 3. The robot found and maintained a

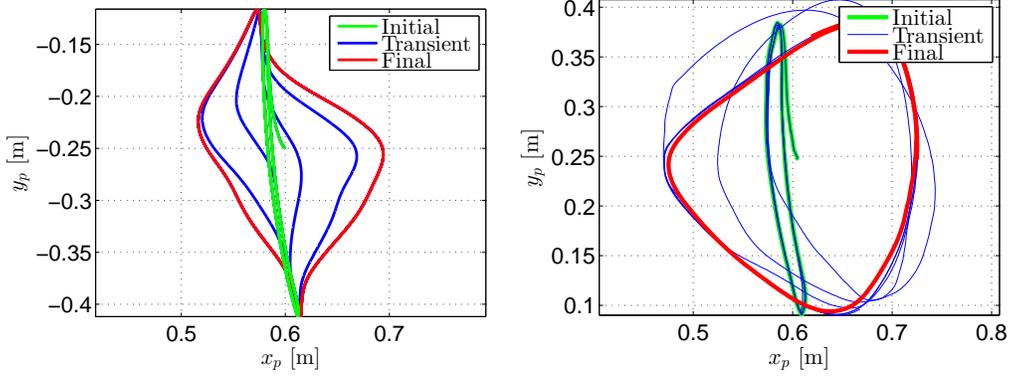


Figure 8: Left: $X - Y$ plot of the end-effector motion depicts the initial motion in green, the motion during coaching in blue, and the final trajectory in red. Right: the same result that led to a different trajectory. The approach of (32) was used.

contact with a flat surface from the start of the experiment. Coaching was applied in x direction only.

Instead of forces as the error of learning as defined in (26), we can also use the position of the robot. By using impedance control mode for the robot and setting a lower stiffness in the direction we want to coach the robot, for example x_p , it will move compliantly in that direction if pushed/pulled. We can now use the difference between the desired and the actual position as the error signal for learning,

$$e_j = x_{p,\text{des}} - x_{p,\text{act}}. \quad (32)$$

The results are shown in the right plot of Fig. 8. It should be noted that when the robot is compliant, contact forces with the surface might affect its trajectory. While this might be solved by changing the stiffness values during coaching and during pure motion reproduction, in our experiment we kept the stiffness constant. Modifying the stiffness is simply a matter of the interface.

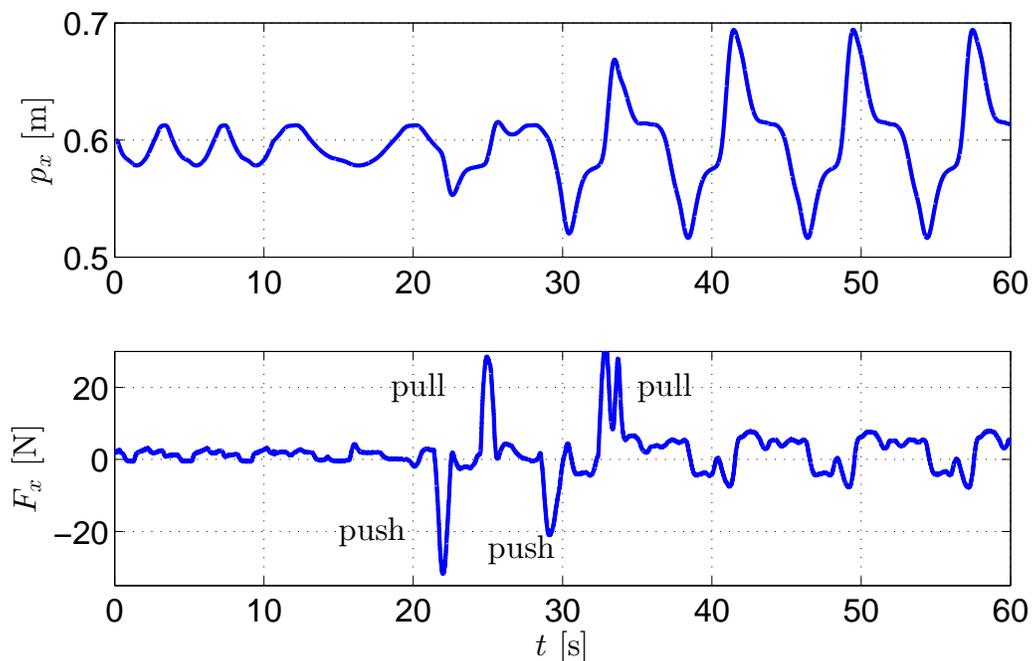


Figure 9: The top plot depicts the trajectory of motion in x_p direction of the end effector of the robot. The external forces applied by the tutor, depicted in the bottom plot, modified the motion to achieve the intended result. The proposed method provides smooth transition from non-coaching to coaching behavior.

The difference of plots in Fig. 8 comes from the compliance of the robots. If the feedback term $d(F)$ in (23) was set higher, the robot would give way much more, and the same principle as in (32) could be applied. We observed that coaching became much more intuitive when the robot was compliant.

When using gestures, we used pointing gestures to coach the robot as defined in (28). We implemented this form of coaching on the JST-ICORP/SARCOS humanoid robot CBi [45]. We used the Microsoft Kinect sensor and the associated body tracker to capture human coaching gestures. Fig. 1 shows the experimental

setup, where the body tracking results can be seen on the display in the background.

To make coaching intuitive, the interface was set so that the human coach can modify the trajectory by either pushing it away from him using his right hand or pulling it towards him with his left hand. The coaching direction was calculated using the wrist and the elbow location. For the right hand, i. e. pushing the trajectory away from the coach, the direction is given by

$$\mathbf{d}_R = \frac{\mathbf{x}_{w,R} - \mathbf{x}_{e,R}}{\|\mathbf{x}_{w,R} - \mathbf{x}_{e,R}\|}, \quad (33)$$

where the $\mathbf{x}_{w,R}$ and the $\mathbf{x}_{e,R}$ are the Cartesian positions of the right hand wrist and the right hand elbow in the robot's base coordinate system. For pulling the trajectory, the direction is given by

$$\mathbf{d}_L = -\frac{\mathbf{x}_{w,L} - \mathbf{x}_{e,L}}{\|\mathbf{x}_{w,L} - \mathbf{x}_{e,L}\|}. \quad (34)$$

Here $\mathbf{x}_{w,L}$ and the $\mathbf{x}_{e,L}$ are respectively the Cartesian positions of the left hand wrist and the left hand elbow in the robot's base coordinate system.

The center of the potential field generated by each hand was moved slightly away from the respective hand. For the right hand, the origin of the potential field defined by the coaching gesture was moved in the direction of the coaching gesture

$$\mathbf{o}_R = \mathbf{x}_R + \xi_R \mathbf{d}_R, \quad (35)$$

where ξ_R is the scalar that defines the distance between the hand and the center of the coaching point in the direction of \mathbf{d}_R . Similar equation is used also for the left hand which attracts the trajectory towards the hand.

$$\mathbf{o}_L = \mathbf{x}_L - \xi_L \mathbf{d}_L. \quad (36)$$

Here, the effective coaching point is moved in the opposite direction of perturbation \mathbf{d}_L . With such modifications the effective origins of potential fields are always in front of the human hands in the direction of pointing at the distance defined by ξ_R and ξ_L .

To determine which hand is active, we use the distance between both wrist positions $\mathbf{x}_{w,L}$, $\mathbf{x}_{w,R}$ and the robot's end-effector position x_p . The active hand is the one which is closer to the robot's hand position.

To show the applicability of the interface for online modification of the initial rhythmic movement using human in the loop coaching gestures, we first provide an example of pulling-in the task space trajectory. The parameters were set to $\gamma = 10$, $\eta = 10$, $r_m = 0.15$ and $\beta = -10/\pi$.

The adaptation is not limited to task space. To update the trajectories in joint-space when they are perturbed in task space, with the coupling term denoted by \mathbf{C}_y , a pseudo inverse of the task Jacobian is used. This essentially maps the task space velocities into the joint space velocities with $\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}}$. By applying a similar transformation to \mathbf{C}_y we obtain

$$\mathbf{C}_q = \mathbf{J}^\dagger \mathbf{C}_y. \quad (37)$$

where $\mathbf{C}_q = [C_{q,1} \ C_{q,2} \ \dots \ C_{q,j}]^T$ and j is the number of the robot's degrees of freedom. The components of (37) are now used for updating the DMP weights w_i using (14) and (13). In this way we ensure that the joint space trajectories encoded by the DMPs are properly modified according to the coach's instructions.

Keeping the movement representation in the joint space is beneficial because our initial movement trajectories, which are encoded by DMPs, are usually acquired by kinesthetic guiding. By using joint space trajectories we avoid losing information about the selected robot configuration during human guiding on a

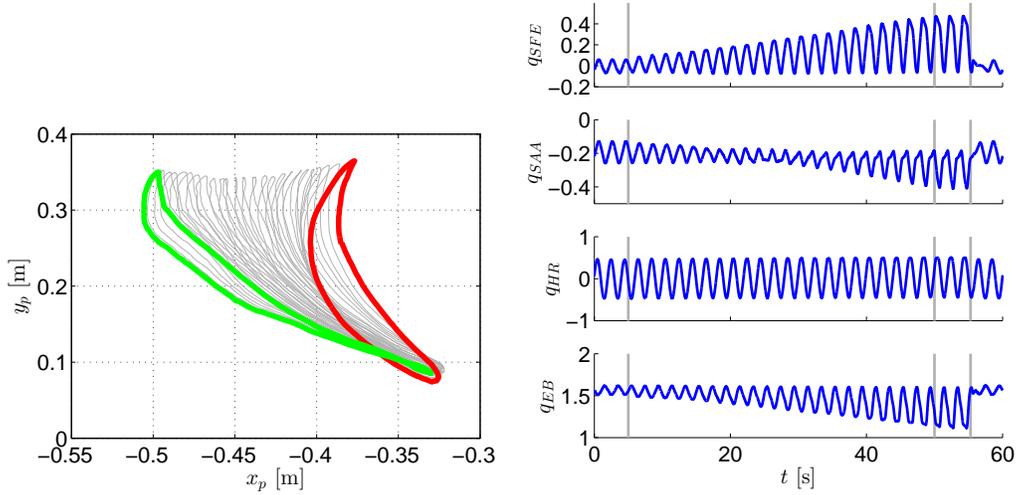


Figure 10: Left: Task space motion of the robot's end-effector, where human coach was modifying the motion pattern. The initial trajectory is in red and the final trajectory is in green. The time evolution of the trajectory modification is indicated with grey line. Right: Joint space motion in time of the robot's right hand, while coaching. Vertical lines indicate the important events described in text.

redundant robot.

Fig. 10-left shows the task space motion of the robot's end-effector in the (x_p, y_p) plane. We can see a successful modification of the motion based on the human coaching gestures. In Fig. 10-right we show the corresponding joint space trajectories as a function of time. The teaching of the new motion pattern begins after 5 seconds, indicated with the first vertical line. The joint space trajectory was modified successfully to achieve the desired task space motion. In Fig. 10-right we can see that at approximately 50 seconds the human coach stopped modifying the behavior and at approximately 55 seconds the new motion pattern was switched back to the original motion pattern. At this point the difference between original motion trajectory and the modified motion trajectory is even more evident.

5.4. *Human-Robot Interface Expansions*

In this section we demonstrate the features of the complete system using an advanced human-robot interface. The system allows the initial transfer of motion, the adaptation to the environment and coaching based on predefined gestures and force interaction. It has been implemented on the humanoid robot ARMAR-IIIa and is used to train periodic DMPs for a wiping task in an online manner. Initially, a DMP is learned from a human wiping movement which is demonstrated in a predefined work space.

Given the color of the wiping tool, the robot tracks the movements of the tool using the stereo camera system of its active head. For the subsequent force-based adaptation of the learned DMP, we rely on the readings of the force torque sensor installed at the humanoid's wrist.

Using the implemented human-robot interface, the human coach can change the learned and adapted DMP using hand gestures. For the recognition of human hand gestures, the robot visually observes the predefined work space in order to localize and track the fingertips of the coaching human hand. To do so, a fingertip tracking algorithm is used which has been introduced in [46]. The fingertips are described with regard to the principal axes spanned by an ellipsoid which circumscribes the entire hand area. Based on these positions a feature vector for the representation of hand gestures was derived. To recognize a gesture, the feature vector is compared with labeled examples which represent a certain gesture. For each coaching mode we defined a distinctive gesture: 1-finger pointing, 2-finger pointing, a fist, and an open hand gesture. The different gestures are depicted in Fig. 11. The 1-finger pointing gesture generates a new target position which is used to change the goal g of the DMP and thus the center of the wiping motion.

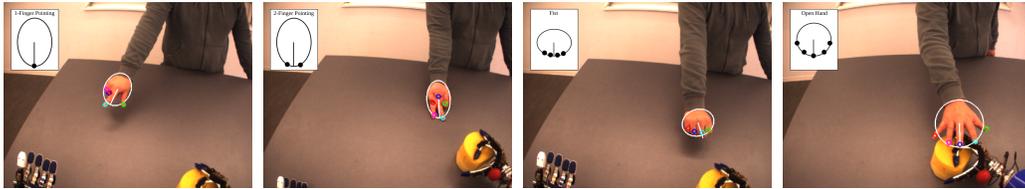


Figure 11: The four hand gesture for the coaching: 1-finger pointing, 2-finger pointing, a fist, and an open hand

The results of changing the center of the wiping motion are shown in Fig. 12.

In order to change the periodic pattern of the wiping motion, the two-finger pointing gesture is used to pull the movements of the robot towards the coaching hand. In contrast, a repelling behavior is triggered using the fist which pushes the robots end-effector away from the human hand. The pushing and pulling behaviours are generated by a virtual potential field imposed on the position of the human hand and, thus, creating virtual forces for the coaching of the DMP. An open hand denotes the approach movement of the coaching hand and invokes a reduction of the frequency with which the wiping motion is reproduced. This facilitates the coaching for the human and allows a smooth transition from vision-based to force-based coaching. The system switches to force-based coaching once an external force is applied on the robots wrist. Using the force-torque sensor, we can coach and further adapt the DMP. During the force-based coaching, the active head shifts its view towards the end-effector. The systems returns to the vision-based coaching once the human hand leaves the currently observed work space. Figure 13 depicts coaching of the ARMAR-IIIa humanoid robot with predefined gestures.

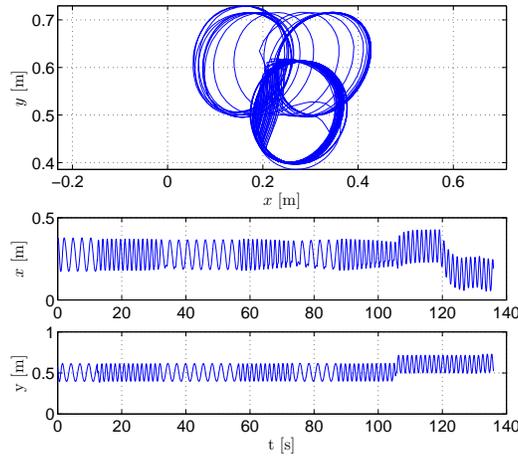


Figure 12: $(x_p - y_p)$ plane trajectories of wiping and coaching on the ARMAR-IIIa robot in the top plot. We can see that the center of the circular trajectory was changed through the coaching interface. Separate directions of motion are depicted in the lower plots. Reduction of the frequency during wiping can be observed in the bottom plots.

6. Discussion and Conclusion

The main advantage of learning the motion required for sustaining a contact is that it allows the combination of feedback and feed-forward control loops. While this by itself is nothing new, the novelty stems from the fact that the feed-forward component is autonomously *learned* and encoded in a dynamic movement primitive. By using the feed-forward component, the feedback component is greatly reduced if not completely canceled, making the behavior exactly as desired. By exploiting the DMP learning mechanisms, we remain in the framework which allows easy modulation with only changing a small set of parameters. Mitigating the need to create models beforehand, as they are learned through exploration and coaching, allows non-experts to effectively transfer motion to the robot by demonstrating everyday tasks.

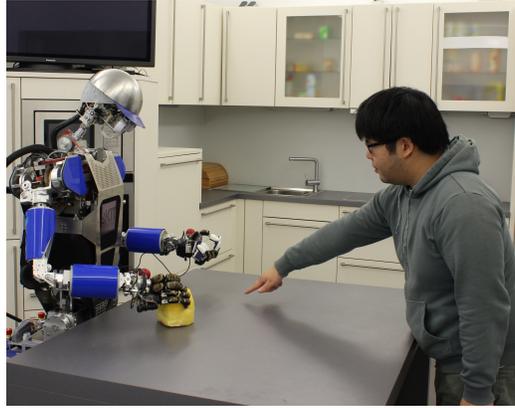


Figure 13: Coaching of the ARMAR-IIIa humanoid robot through the use of hand gestures.

In comparison to other approaches, several sub-areas of research need to be considered. Force control has been applied in robotics in many different contexts. The benefit of our method is that it allows easy and intuitive transfer of motion from a human to the robot. This transferred motion adapts to the conditions of the task – for example, that it needs to maintain contact with the environment. As stated in the introduction, various techniques exist for that, but the methods presented in this paper extend this feature to a well developed and extensively applied framework. The approach can be used on position controlled robots, or on torque controlled robots, exploiting the properties of different control methods, such as impedance control as depicted in Fig. 8. On the other hand, it also allows the use of position controlled robots, such as the ARMAR-IIIa, with the only difference in behavior due to lower bandwidths.

For learning, the exploration of the trajectory space uses the learning algorithm of the DMPs, which is computationally light and allows for quick adaptation. In the case of periodic motions, this can be in the rank of a few periods [40]. The method of direct DMP adaptation, which was also applied in the context of

coaching, observes similar principles as iterative learning control, and could be considered an instance of it. It enables direct learning of the weights of DMP kernel functions instead of the signal. Again, remaining in the DMP framework has beneficial properties for robot control.

For motion adaptation by coaching, our approach thus retains the beneficial properties of DMPs with time-invariance and the means of modulation, but additionally enables the modulation of complex motions through intuitive coaching gestures. An advanced but intuitive coaching interface, which was demonstrated on the ARMAR-IIIa robot, has proven to be a viable solution.

Adaptation to the environment, as presented in this paper, exploits the knowledge of the demonstrator to determine the needed references and directions for adaptation. An open research issue remains, how such adaptations can be performed autonomously. While a complex cognitive reasoning system behind this is beyond the scope of this paper, simple conditions could greatly improve the autonomy of adaptation. For example, in the context of wiping, we could direct the robot to increase the force of contact with the surface in case the wiping does not actually remove the identified dirt. Vision systems, specifically using RGB-D sensors, are also efficient at detecting surfaces. These surfaces could be the targets of wiping when a wiping command is issued. Any such augmentation of the interface can greatly improve the user experience.

In the paper we proposed and evaluated several methods for adaptation of DMPs based on force feedback. We have shown that all can be effectively used for acquiring and maintaining non-rigid contacts with the environment. They thus offer a viable solution for an inclusion in future household assistants. In the future we will combine the method of DMP adaptation, effectively applied to coaching,

to modify feed-forward models of complex tasks. For example, one might update the demonstrated DMP so that the postural stability of the robot is observed. Another possible application of our approach is the learning of the required torque signals for robot control.

Acknowledgement

This work was supported by the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) grant agreement no. 270273, Xperience. Additionally, A.U. was supported by NICT, Japan Trust International Research Cooperation Program; J.M. was supported by MEXT KAKENHI 23120004; by MEXT SRPBS; by MIC-SCOPE; by JST-SICP; by ImPACT Program of Council for Science, Technology and Innovation.

- [1] R. Dillmann, Teaching and learning of robot tasks via observation of human performance, *Robotics and Autonomous Systems* 47 (2-3) (2004) 109–116.
- [2] A. Ude, A. Gams, T. Asfour, J. Morimoto, Task-specific generalization of discrete and periodic dynamic movement primitives, *IEEE Transactions on Robotics* 26 (5) (2010) 800–815.
- [3] A. Ude, C. G. Atkeson, M. Riley, Programming full-body movements for humanoid robots by observation, *Robotics and Autonomous Systems* 47 (2-3) (2004) 93–108.
- [4] Y. Wada, M. Kawato, A via-point time optimization algorithm for complex sequential trajectory formation, *Neural Networks* 17 (3) (2004) 353–364.

- [5] S. Calinon, F. D’halluin, E. L. Sauser, D. G. Caldwell, A. G. Billard, Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression, *IEEE Robotics and Automation Magazine* 17 (2) (2010) 44–54.
- [6] S. Khansari-Zadeh, A. Billard, Imitation learning of globally stable nonlinear point-to-point robot motions using nonlinear programming, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010, pp. 2676–2683.
- [7] E. Gribovskaya, S. Khansari-Zadeh, A. Billard, Learning non-linear multivariate dynamics of motion in robotic manipulators, *International Journal of Robotics Research* 30 (1) (2011) 80–117.
- [8] T. Inamura, I. Toshima, H. Tanie, Y. Nakamura, Embodied symbol emergence based on mimesis theory, *International Journal of Robotics Research* 23 (4-5) (2004) 363–377.
- [9] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, S. Schaal, Dynamical movement primitives: Learning attractor models for motor behaviors, *Neural Computation* 25 (2) (2013) 328–373.
- [10] T. Matsubara, S.-H. Hyon, J. Morimoto, Learning parametric dynamic movement primitives from multiple demonstrations, *Neural Networks* 24 (5) (2011) 493–500.
- [11] T. Kulvicius, K. Ning, M. Tamosiunaite, F. Wörgötter, Joining movement sequences: Modified dynamic movement primitives for robotics applications

- exemplified on handwriting, *IEEE Transactions on Robotics* 28 (1) (2012) 145–157.
- [12] B. Nemeč, A. Ude, Action sequencing using dynamic movement primitives, *Robotica* 30 (5) (2012) 837–846.
- [13] J. Peters, S. Schaal, Reinforcement learning of motor skills with policy gradients, *Neural Networks* 21 (2008) 682–697.
- [14] F. Stulp, E. A. Theodorou, S. Schaal, Reinforcement learning with sequences of motion primitives for robust manipulation, *IEEE Transactions on Robotics* 28 (6) (2012) 1360–1370.
- [15] J. Kober, A. Wilhelm, E. Oztop, J. Peters, Reinforcement learning to adjust parametrized motor primitives to new situations, *Autonomous Robots* 33 (2012) 361–379.
- [16] M. Tamosiunaite, B. Nemeč, A. Ude, F. Wörgötter, Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives, *Robotics and Autonomous Systems* 59 (11) (2011) 910–922.
- [17] L. Villani, J. De Schutter, Force control, in: B. Siciliano, O. Khatib (Eds.), *Handbook of Robotics*, Springer, 2008, pp. 161–185.
- [18] N. Hogan, Impedance control: An approach to manipulation I II III, *Journal of Dynamic Systems, Measurement and Control* 107 (1985) 1–24.
- [19] V. Koropouli, D. Lee, S. Hirche, Learning interaction control policies by demonstration, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 344–349.

- [20] P. Pastor, L. Righetti, M. Kalakrishnan, S. Schaal, Online movement adaptation based on previous sensor experiences, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, 2011, pp. 365–371.
- [21] P. Pastor, M. Kalakrishnan, L. Righetti, S. Schaal, Towards associative skill memories, in: IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2012, pp. 309–315.
- [22] M. Kalakrishnan, L. Righetti, P. Pastor, S. Schaal, Learning force control policies for compliant manipulation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011, pp. 4639–4644.
- [23] T. Kulvicius, M. Biehl, M. J. Aein, M. Tamosiunaite, F. Wörgötter, Interaction learning for dynamic movement primitives used in cooperative robotic tasks, *Robotics and Autonomous Systems* 61 (12) (2013) 1450 – 1459.
- [24] A. Gams, B. Nemeč, A. Ijspeert, A. Ude, Coupling movement primitives: Interaction with the environment and bimanual tasks, *IEEE Transactions on Robotics* 30 (4) (2014) 816–830.
- [25] E. L. Sauser, B. Argall, G. Metta, A. Billard, Iterative learning of grasp adaptation through human corrections, *Robotics and Autonomous Systems* 60 (1) (2012) 55–71.
- [26] S. Calinon, A. Billard, Incremental learning of gestures by imitation in a humanoid robot, in: Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2007, pp. 255–262.

- [27] F. J. Abu-Dakka, B. Nemeč, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, A. Ude, Adaptation of manipulation skills in physical contact with the environment to reference force profiles, *Autonomous Robots* 39 (2) (2015) 199–217.
- [28] L. Rozo, P. Jimnez, C. Torras, A robot learning from demonstration framework to perform force-based manipulation tasks, *Intelligent Service Robotics* 6 (1) (2013) 33–51.
- [29] L. Rozo, P. Jimenez, C. Torras, Force-based robot learning of pouring skills using parametric hidden markov models, in: *Workshop on Robot Motion and Control (RoMoCo)*, 2013, pp. 227–232.
- [30] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, D. G. Caldwell, Statistical dynamical systems for skills acquisition in humanoids, in: *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 323–329.
- [31] A. Gams, M. Do, A. Ude, T. Asfour, R. Dillmann, On-line periodic movement and force-profile learning for adaptation to new surfaces, in: *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Nashville, TN, 2010, pp. 560–565.
- [32] A. Gams, T. Petrič, B. Nemeč, A. Ude, Learning and adaptation of periodic motion primitives based on force feedback and human coaching interaction, in: *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 166 – 171.
- [33] J. Ernesti, L. Righetti, M. Do, T. Asfour, S. Schaal, Encoding of periodic and their transient motions by a single dynamic movement primitive, in: *IEEE-*

- RAS International Conference on Humanoid Robots (Humanoids), 2012, pp. 57–64.
- [34] M. Do, J. Schill, J. Ernesti, T. Asfour, Learn to wipe: A case study of structural bootstrapping from sensorimotor experience, in: IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1858 – 1864.
- [35] v. Ortenzi, M. Adjigble, J. A. Kuo, R. Stolkin, M. Mistry, An experimental study of robot control during environmental contacts based on projected operational space dynamics, in: IEEE-RAS International Conference on Humanoid Robots, 2014, pp. 407 – 412.
- [36] A. Gruebler, V. Berenz, K. Suzuki, Coaching robot behavior using continuous physiological affective feedback, in: IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2011, pp. 466–471.
- [37] M. Riley, A. Ude, C. Atkeson, G. Cheng, Coaching: An approach to efficiently and intuitively create humanoid robot behaviors, in: IEEE-RAS International Conference on Humanoid Robots, 2006, pp. 567–574.
- [38] D. Lee, C. Ott, Incremental kinesthetic teaching of motion primitives using the motion refinement tube, *Autonomous Robots* 31 (2-3) (2011) 115–131. doi:10.1007/s10514-011-9234-3.
- [39] T. Petrič, A. Gams, L. Žlajpah, A. Ude, J. Morimoto, Online approach for altering robot behaviors based on human in the loop coaching gestures, in: IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 4770 – 4776.

- [40] A. Gams, A. J. Ijspeert, S. Schaal, J. Lenarčič, On-line learning and modulation of periodic movements with nonlinear dynamical systems, *Autonomous Robots* 27 (1) (2009) 3–23.
- [41] T. Petrič, A. Gams, A. J. Ijspeert, L. Žlajpah, On-line frequency adaptation and movement imitation for rhythmic robotic tasks, *The International Journal of Robotics Research* 30 (14) (2011) 1775–1788.
- [42] C. G. Atkeson, A. W. Moore, S. Schaal, Locally weighted learning, *AI Review* 11 (1997) 11–73.
- [43] L. Ljung, T. Söderström, *Theory and Practice of Recursive Identification*, MIT Press, 1986.
- [44] H. Hoffmann, P. Pastor, D.-H. Park, S. Schaal, Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance, in: *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009, pp. 2587–2592.
- [45] G. Cheng, S.-H. Hyon, J. Morimoto, A. Ude, J. G. Hale, G. Colvin, W. Scroggin, S. C. Jacobsen, CB: A humanoid research platform for exploring neuroscience, *Advanced Robotics* 21 (10) (2007) 1097–1114.
- [46] M. Do, T. Asfour, R. Dillman, Particle filter-based fingertip tracking with circular hough transform features, in: *Proceedings of the 12th IAPR Conference on Machine Vision Application*, 2011, pp. 1 – 4.